Methodology

Introduction about data

- Residential houses gathered by TU Delft project OPSCHALER (1-hour gas resolution, 54/61 houses). Sampling period varies from 3 weeks to 8 months.
- Weather data from KNMI Rotterdam (15 min resolution)

X.1. One-time step ahead forecasting

In one-time step ahead forecasting, a distinction is made between an hourly and daily resolution as shown in Table X. Hourly forecasting is useful for short term insights in gas usage, while daily forecasting is useful for creating a more general model of gas usage. As seen in the table below the data down sampled to a daily resolution performs better on MAPE and SMAPE in comparison to hourly resolution.

Data cleaning

- Combining data sets
- Removing NANs
- Resampling

Model structure explanation

Data processing:

- Data selection
- Feature scaling
- Data splitting
- Dummy variables

Models

MVLR

Multivariate Linear Regression (MVLR) is the simplest model used for the prediction of gas consumption, and it gives a general view of data. MVLR is based on model response variables (*y1, y2…, yi*) as a function of predictor variables (*x1, x2, …, xk)*.

The first step of MVLR is to decide which features are dropped and which features are used as predictor variables for creating the function, a helpful way for choosing them is looking into the interaction and correlation coefficient between each other. Secondly, each feature is an independent variable multiplied by an adjusted weight during the training set known as *regression coefficients*. The result of the summation of all these parameters is the prediction.

Sigma is the residual term of the model.

DNN (feedforward)

A feedforward neural network is one of the simplest type of artificial neural network due to that fact all the connections go in one direction, without any cycles. The data from the input nodes is passed on to the next layers of neurons (hidden layers) and based on the logistic equation they compute and pass something to next layers until the output node is reached.

In contrast to MVLR, a feedforward neural network automatically determines the correct weights to fit the data. It can thus be seen as a black box model. Depending on the goal of the forecasting, this might be an advantage or disadvantage. For example, if the goal is to get insight into building characteristics, it will be difficult to get this information extracted.

GRU

As stated a GRU model is a type of RNN model. GRU's can be compared to a more efficient and simplified way of the LSTM model. To solve the gradient problem of a standard RNN, a GRU uses a reset and update gate. A GRU also has no cell state and uses the hidden state to transfer the 'Opschaler data'.

As mentioned above a GRU only has two gates: a update gate and a reset gate. These vectors decide what information is going to be passed to the output. A GRU is 'special' because it can be trained to keep information from a long time ago without removing relevant information. This makes the GRU model more suitable for predicational methods.

The update gate helps determine how many of the previous information it needs to make a prediction. That is really positive because the GRU can decide to copy all the information from the past steps and eliminate the risk of vanishing gradient problem. The reset gate however is a gate that decides how much past steps it can forget. In short the advantages entail:

- The GRU is faster than an LSTM due to fewer tensor operations.
- The GRU has total control over the flow of information without using a memory unit.
- The GRU model works more efficiently compared to an LSTM model, due to its more efficient use of gates.

Disadvantages

- An LSTM can use more parameters than a GRU
- Similarly there is issue of increasing gradients at each step called as exploding gradients

Complex models

Loss functions / Cost functions

TODO: Add proper references

The Mean Squared Error (MSE) is used as the loss function for all the models and is defined as follows

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

Where $\widehat{Y_i}$ is the i-th ground truth value, Y_i is the i-th predicted value and n is the total number of samples.

In general the MSE is easy to solve and it is sensitive to outliers because the difference between Y_i and $\widehat{Y_i}$ is squared. This sensitivity to outliers has a positive influence for training the models on the used dataset. This is because the outliers in the gas consumption represent valid data points, e.g. they are not corrupt data due to malfunctioning of the measurement devices.

The main reason to use the MSE instead of the Mean Absolute Percentage (MAPE) or the Symmetric Mean Absolute Percentage Error (SMAPE) is because MAPE and SMAPE minimize less at the geometric centre of a distribution in comparison to MSE. The MAPE is defined as follows

Met opmerkingen [BK1]: Add more reasons maybe?

Met opmerkingen [BK2]: Meaning that they are less sensitive for outliers

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|\widehat{Y_\iota} - Y_i|}{|Y_i|}$$

and SMAPE as

$$\text{SMAPE} = \frac{100\%}{2n}\sum_{i=1}^{n}\frac{|Y_i - \widehat{Y_\iota}|}{|\widehat{Y_\iota}| + |\widehat{Y_\iota}|}.$$

Notice how the MSE is scale dependant whereas MAPE and SMAPE are in percentages. The MAPE and SMAPE are used as evaluation metrics, together with MSE to determine the performance of each model.

Met opmerkingen [BK3]: Maybe say something about MSE getting very low real quick, when the scale of the Y data isn't 'large', because of the small differences (number wise) between predictions and ground truth.

NOTES:

Article on loss functions: https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0

Optimizers

Met opmerkingen [BK4]: Specifying the mathematics is too much.

To minimize the loss function MSE, Adam and Nadam are used as optimizers in the models. In chapter …. is specified which model uses which optimizer. The main difference between Adam and Nadam is that Adam is essentially RMSprop with momenten whereas Nadam is Adam RMSprop with Nesterov momentum. Putting this in easy terms, Nadam in $\mathbb{R}^3$ will jump over hills quicker than Adam does by default. Whereas Adam converges quicker than stochastic gradient decent.

Met opmerkingen [BK5]: Or cost function

Normalization

The features X are standardized by removing the mean and scaling to unit variance using the StandardScaler from scikit-learn. This transformation makes it so all features are on the same scale. Scaling the features is to prevent one feature dominating another due to a relatively large difference in the feature scales. The models used will also converge less quick and have a likelihood to have a lower accuracy when the features are not scaled.

From the train set, each feature is scaled independently. The standard score Z of the feature sample is

$$Z = \frac{X - u}{s}.$$

Met opmerkingen [BK6]: This might be left out, dunno

Where X is the feature sample, u is the mean and s is the standard deviation from the sample. Per feature this information is stored and applied is also to the test features, to transform them in the same way as the train data without having data leakage of the distribution of the feature samples from the test set.

Results – Opschaler

X. Results and discussion

A field test has been deployed in order to evaluate the forecast accuracy of the proposed machine learning algorithms for the prediction of gas usage in residential homes The results of the field test are presented in Table 1. Each algorithm is evaluated making use of five criteria: MSE, MPAE %, SMAPE , time per epoch and epochs required. The parameters used to make the prediction with each of the algorithms and the characteristics of the evaluation criteria has been presented in the methodology chapter.

Table 1. Performance of machine learning algorithms applied for residential gas usage predictions.

X.2. Simplistic prediction models: Multivariate linear regression model(MVLR) and Deep neural network(DNN)

Met opmerkingen [L7]: Only the first time you present the name of the algorithm you will use the full name followed of the abbreviation between round brackets (). After that you will only use the abbreviation.

The accuracy of the MVLR algorithm with hourly resolution, MAPE 2362.7%, differs completely from the accuracy with daily resolution, MAPE 24%.. The difference is cause by outliers, this is due tto he differences I the seasons.

Met opmerkingen [L8]: You can explain better why there is such a big difference between the two of them. What causes the outliers in hour resolution, and why you do not have outliers in day resolution?

DNN MAPE% for daily resolution is 57,1% what is largely better than the MAPE 2362.7% of MVLR. The reason that DNN performs better than MVLR with daily resolution is, this has to do with the memory gates which remembers the previous data on which it has a better performance

On the other hand MVLR outperforms DNN by a 9,4% on MAPE with daily resolution.

Met opmerkingen [L9]: Why is MVLR performing better than MVLR in hourly resolution.

Model	Resolution	MSE	MAPE (%)	SMAPE (%)	Time per epoch (micro seconds)	Epochs required
MVLR	Hourly		2362.79			
	Daily	96.59	20.4	9.3		
DNN	Hourly	0.7	57.1	19.2	4,0	30
	Daily	147.7	29.8	12.3		
LSTM	Hourly	0.7	51.9	25.5	69,0	150
	Daily	79.4	21.5	10.5		
GRU	Hourly	0.7	46.7	25.9	69,0	110
	Daily	87.6	25.1	13.1		
CNN	Hourly	0.8	55.9	26.6	45	14
	Daily	103.9	26.2	12.7		
"Opschaler"	Hourly				404	50-100*
	Daily		12.5	6.2		

TABLE 1

X.3. Complex neural networks: Long-Short term memory model(LSTM), GRU and Convolutional neural network(CNN)

Table 1. illustrates other prediction models which uses more computational power, but present better results. The LSTM model and the GRU model are as mentioned in the methodology both part of the Recurrent neural network. Based on the TABLE results it can be stated that the prediction performance of the GRU has a tremendous advantage compared to the LSTM and CNN and models.

In comparison to the hourly resolution on the SMAPE and MAPE values certain models exceeds another. CNN has an hourly MAPE resolution of 55.9% and a SMAPE value of 26.6%. GRU has a MAPE value of 46.7% and a SMAPE of 25.9%. LSTM has a MAPE value of 51.9% and a SMAPE value of 25.5%. Based on these values it is noticed that the GRU model has the lowest MAPE value and thus a better performance than the other complex neural network models. Whereas LSTM has the lowest SMAPE value, GRU still has a better overall performance. The noticeable difference on hourly resolution between the SMAPE values of LSTM and GRU is only an 0,4% difference.

In comparison the models based on the daily resolution it is seen(TABLE) that the CNN model with a MAPE of 26.2 and a SMAPE of 12.7% has a disadvantage. Which is reflected in the MAPE and SMAPE values in comparison the LSTM and GRU models. Even though the LSTM model takes advantage of the time series data it still has a disadvantage compared to the GRU model. In reference to the number of epochs and the time it takes to train a model. The GRU is faster to train than a LSTM model, due to its lack of needing memory units, which is reflected in the table and the number of epochs(110).

Met opmerkingen [L10]: This is out of place. I do not undertand what it is doing here.

Met opmerkingen [L11]: I have not corrected this section. Complex neural networks. But it should follow the same structure as the previous section.

You compare first the three algorithms LSTM, GRU and CNN between them on hourly resolution. You tell which of them performs better. You add the number of the main performance criteria used, MAPE, and you explain the why of the difference on the performance. In case needed you add extra information of the performance of the algorithms with the other criteria. Like you did in the last section.

Second you compare the performance of the three algorithms between them with daily resolution. You compare them in the same way you did with the hourly resolution.

Third, you compare in general these three methods, complex neural networks with the simplistic prediction models. You follow the same order first hourly resolution and later daily resolution.

X.4. *Combinational neural networks DNN/RNN/CNN*

By combining the different models, a higher accuracy can be achieved as the SMAPE of the " "Opschaler" model illustrates. With a SMAPE of 6.2%, the model is twice as accurate as the other proposed models. This complex model results in an increase in time per epoch <…> and epochs required <…> to accomplish these accuracy rates. { FIG} contains the values of <drunkschaler>'s loss function per epoch. After this point <enter epochs needed up untill this point> the model no longer converges quickly, whereas after epoch <…> it has reached a global minimum.

Worth noticing is the <…%> difference between the training and validation loss. All models have this visible variance between the train and validation loss. The loss plots from the other models can be found in Appendix <…>.

Met opmerkingen [L12]: This looks better, but still needs to follow the same structure. First hourly resolution, performance values and why.
Second daily resolution, performance values and why.

<Results table>, <drunkschaler loss plot>, <prediction plots (daily resolution?) of all models in one plot?>

Table … contains the cross-validation evaluation metrics of the used models. The models have been cross-validated on the test dataset, as is explained in chapter <…>.

<… text about the performance of MVLR, DNN, RNN made by group 1>.

The performance of the used CNN is in between the performance of the DNN and GRU models. It is visible in Table … that <drunkschaler> is outperforming the second-best model by <…> %. Figure … contains the values of <drunkschaler>s' loss function per epoch. <Drunkschaler> has converged the most after <…20? 25?> epochs. After this point the model no longer converges quickly, whereas after epoch … it has reached a (local) minimum. This is after … minutes of training on a GTX 970 GPU. Notice the … % difference between the training and validation loss. All models have this visible variance between the train and validation loss. The loss plots from the other models can be found in Appendix <…>.

Met opmerkingen [KBd(13]: Which indicates that more training data will improve the accuracy of the model.

Uncertainty

Vertellen hoe de resultaten veranderen bij het aan passen van lookback, train size? Image grootte/ hoeveelheid kleine images?

--- Things to add

CNN sometimes need to be re-run from scratch almost 10 times before getting the 'best' result that's about 51-56% MAPE. This is due to the weights randomly initializing probable and/or due to the start of the solution space for adam and the steps adam takes to the (local) minimum.

conclusion

The decrease of the MAPE and SMAPE on a daily resolution in comparison to the hourly resolution can be clarified due to the decrease in oscillation. The predictions of hourly models are therefore less uncertain, creating a higher accuracy.

According to research, CNN should perform better than DNN [1]. With a 0.4% higher SMAPE, in comparison to the DNN model, CNN performs unsatisfactorily. This can be explained due to the task that is proposed to the CNN network, is no classification task in itself. Changing the data structure, of the available data, to the mandatory data structure used by CNN, might cause the difference in SMAPE.
The results of Opschaler are in line with the expectations discussed in <insert chapter methodology>.
<talking about the advantages of CNN> However, when implemented from scratch and trained with a limited amount of data, results are lower than expected, throwing a 55.9% MAPE in the hourly prediction and requiring training time per epoch of 14 microseconds. It has been tested that larger CNN leads to more accurate predictions, but training time increases to around 2 minutes per epoch. In addition to this, when implementing CNNs the time series nature of data is “ignored” losing valuable information that could lead to more accurate results. To address this problem next model is implemented, obtaining better results in terms of accuracy but requiring more computing time.
<Talking about Opschaler module> It is clear that this model implementation outperforms previous ones significantly in terms of accuracy, but it also requires careful pre-processing of data as explained in the methodology section. Another disadvantage that is worth mentioning is the high computing time required of this model, as it could be expected due to the addition of different NN sequentially.